

FIG. 1A
Prior Art

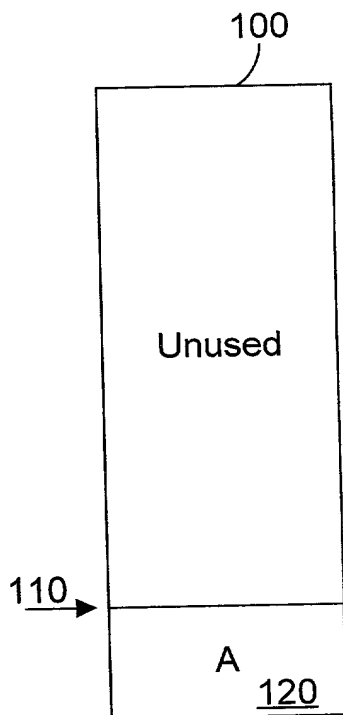


FIG. 1B
Prior Art

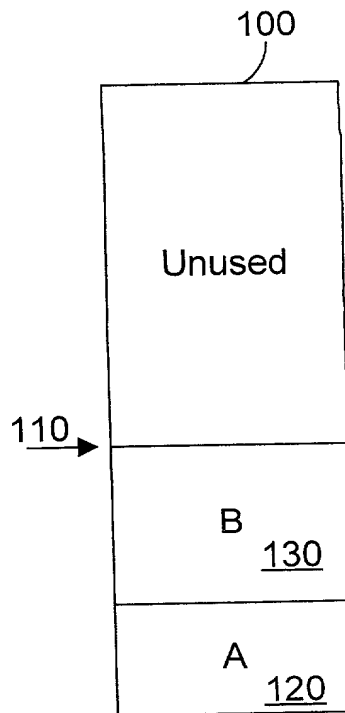


FIG. 1C
Prior Art

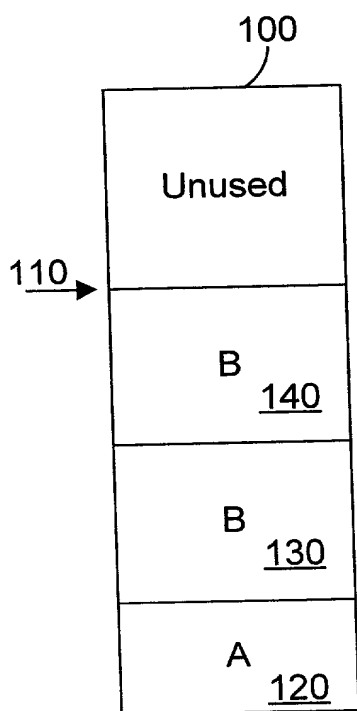


FIG. 1D
Prior Art

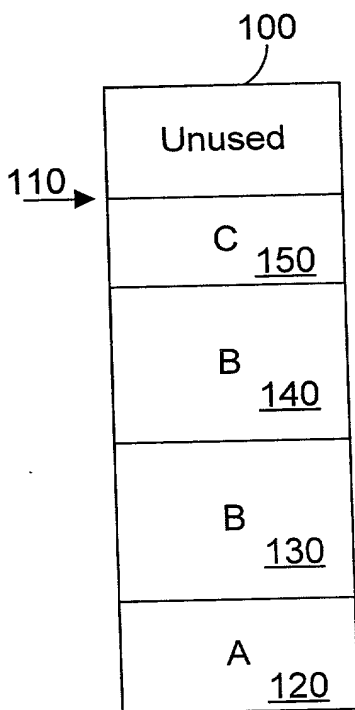


FIG. 1E
Prior Art

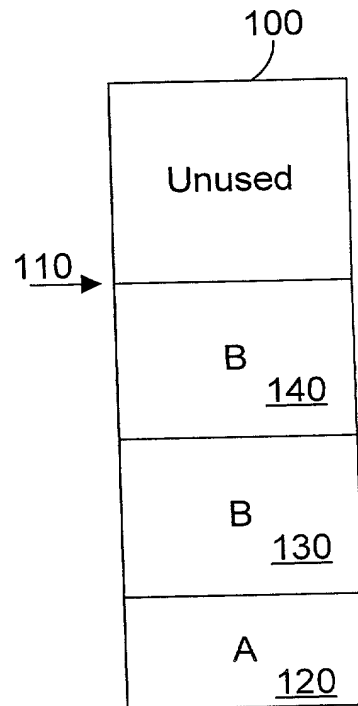


FIG. 1F
Prior Art

FIG. 1A-1F

```
C++  
A() {  
    Square k;  
    ...  
}
```

FIG. 2A Prior Art

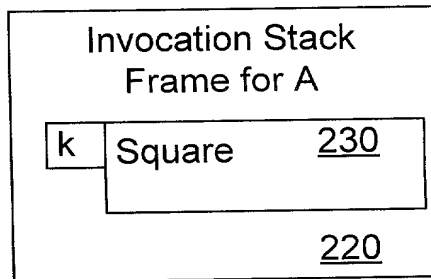


FIG. 2B Prior Art

```
Java  
A() {  
    ...  
    Square k = new Square();  
    ...  
}
```

FIG. 3A Prior Art

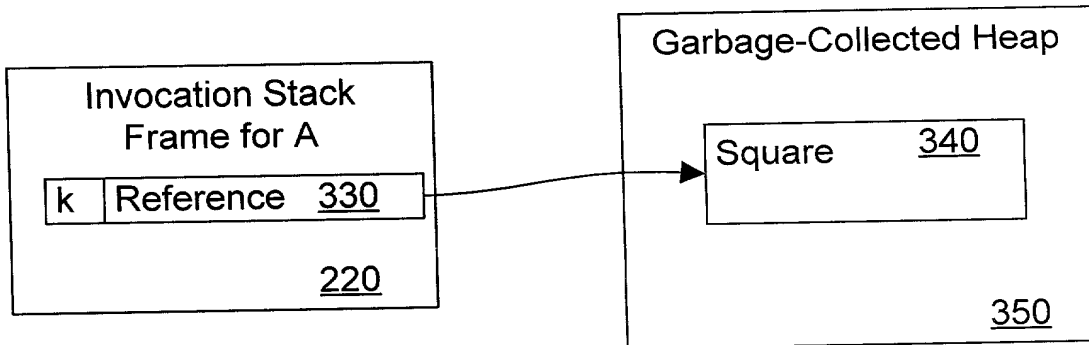


FIG. 3B Prior Art

FIG. 3C Prior Art

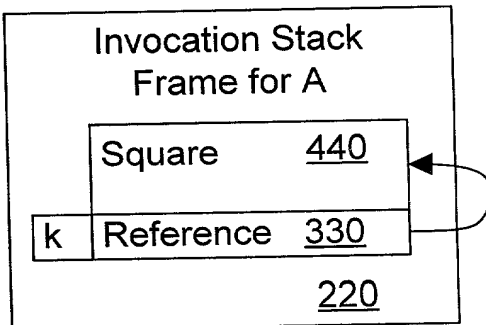


FIG. 4 Prior Art

FIG. 2A FIG. 2B FIG. 3A FIG. 3B FIG. 3C FIG. 4

```
A() { NO ESCAPE  
  ⋮  
  Square k = new Square();  
  ⋮  
}
```

FIG. 5A Prior Art

```
static Square classVar;  
A() { GLOBAL ESCAPE  
  ⋮  
  Square k = new Square();  
  classVar = k;  
  ⋮  
}
```

FIG. 5B Prior Art

```
Square A() { ARG ESCAPE  
  ⋮  
  Square k = new Square();  
  return k;  
  ⋮  
}
```

FIG. 5C Prior Art

```
A(List L) { ARG ESCAPE  
  ⋮  
  Square k = new Square();  
  L.addToList(k);  
  ⋮  
}
```

FIG. 5D Prior Art

FIG. 5A-5D

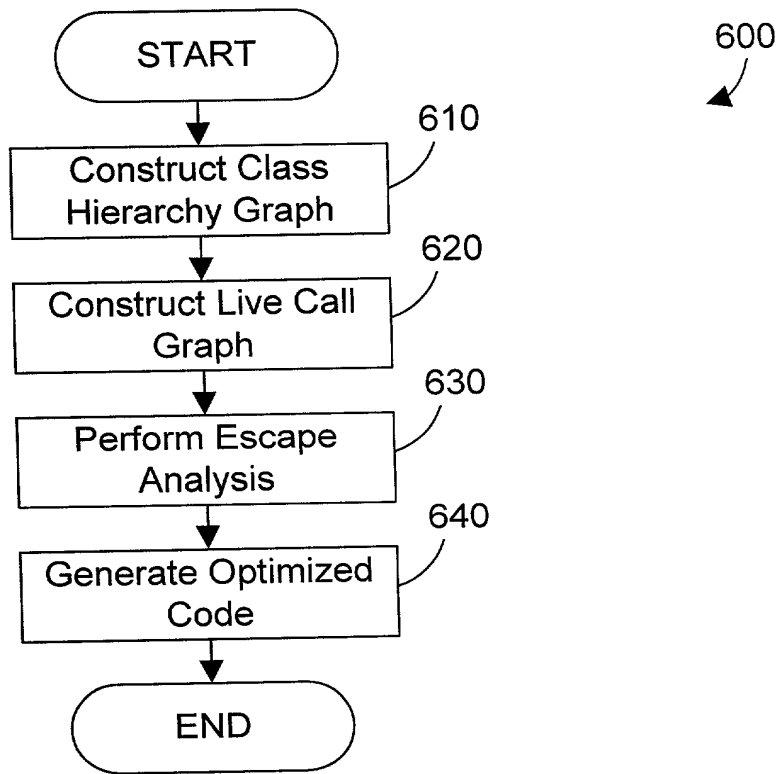


FIG. 6 Prior Art

FIG. 6

630

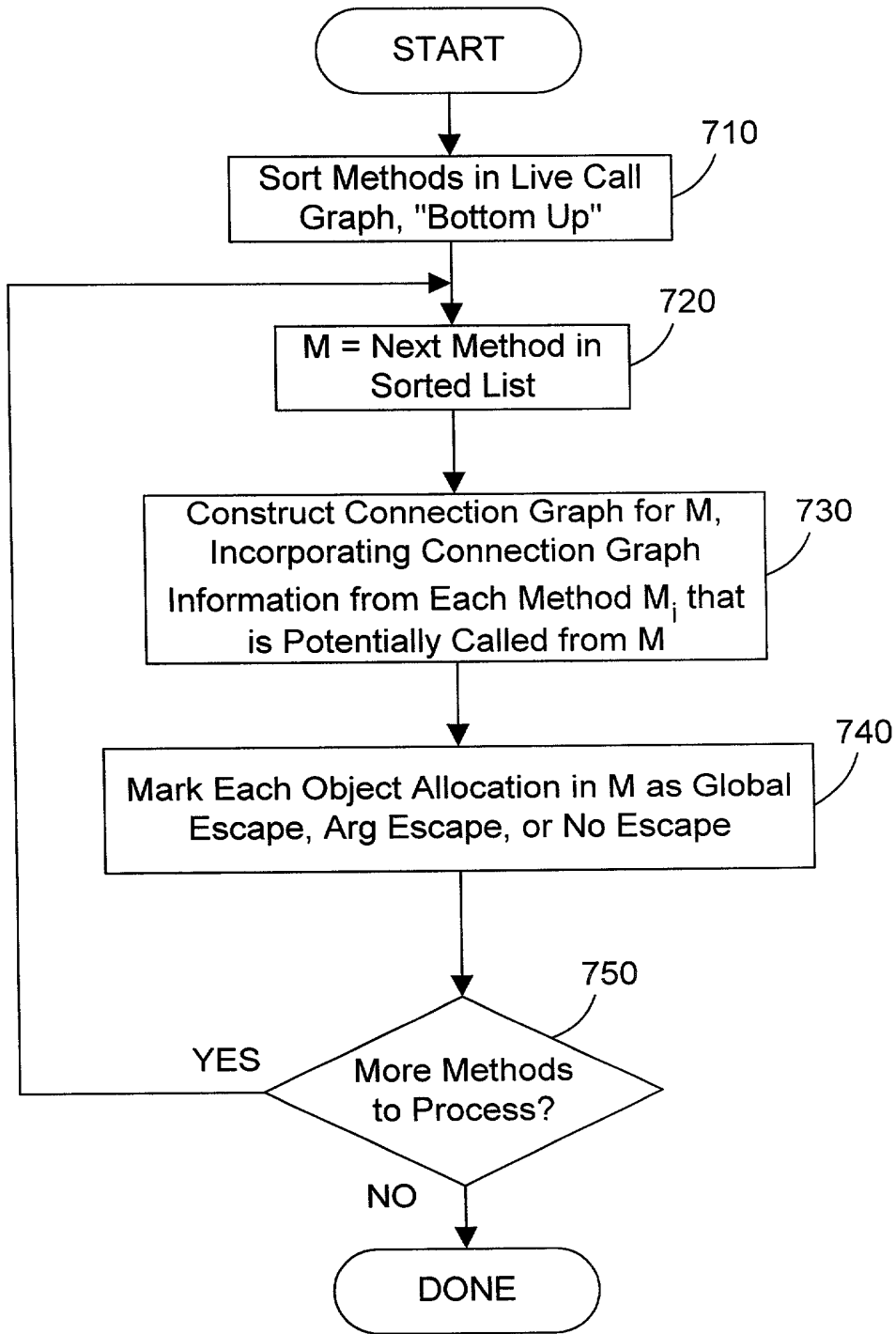


FIG. 7 Prior Art

FIG. 7

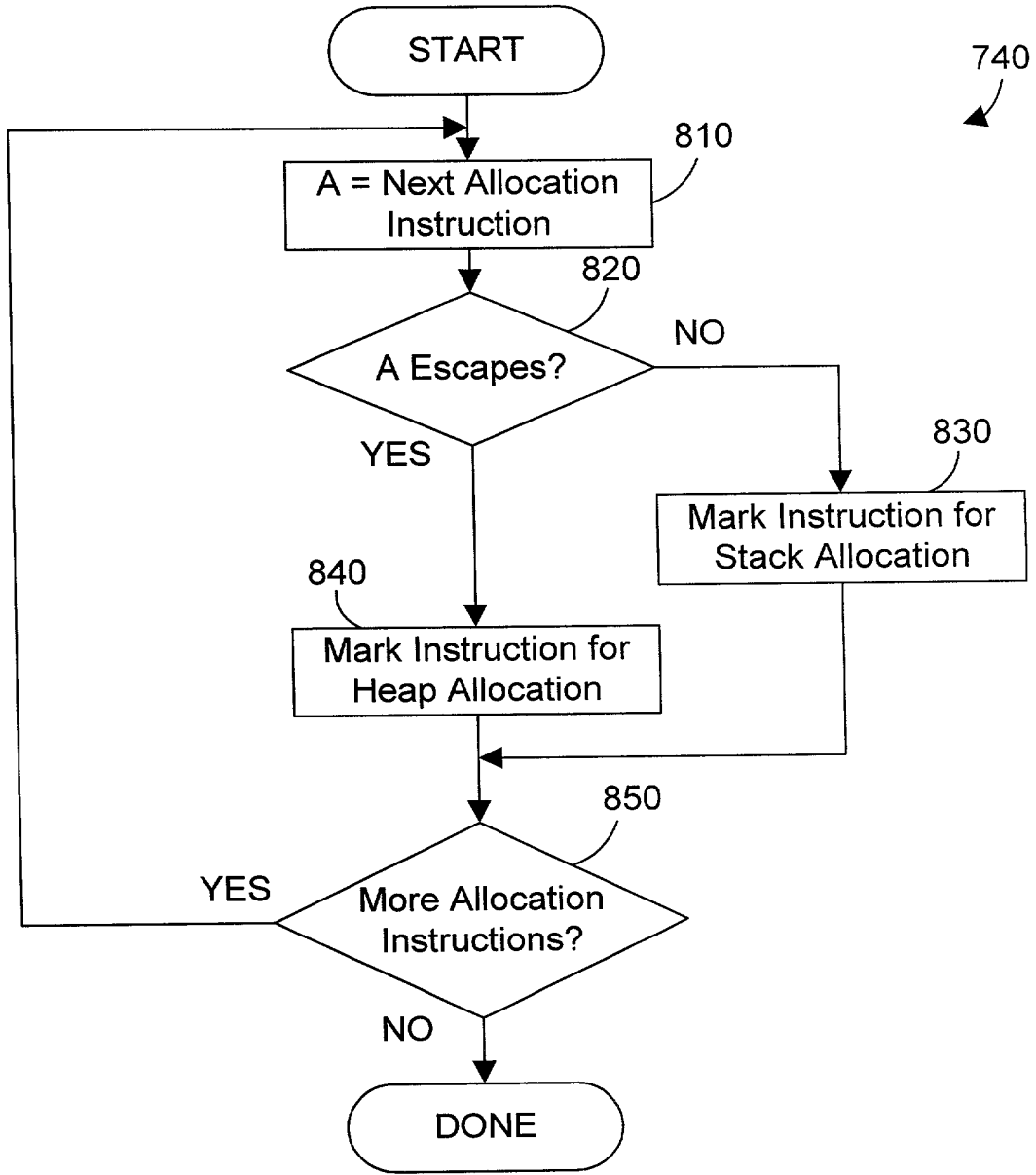


FIG. 8 Prior Art

FIG. 8

900

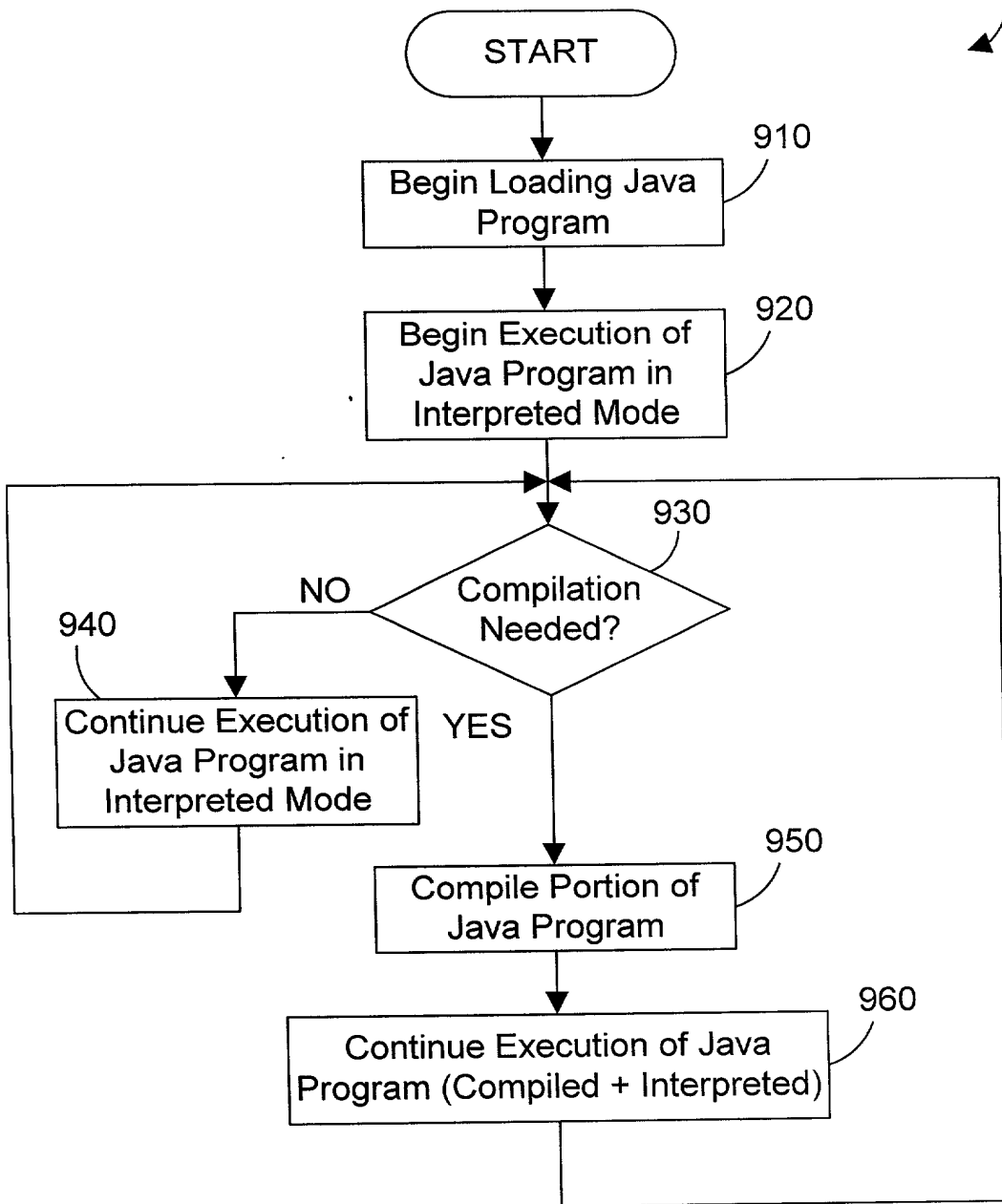


FIG. 9 Prior Art

FIG. 9

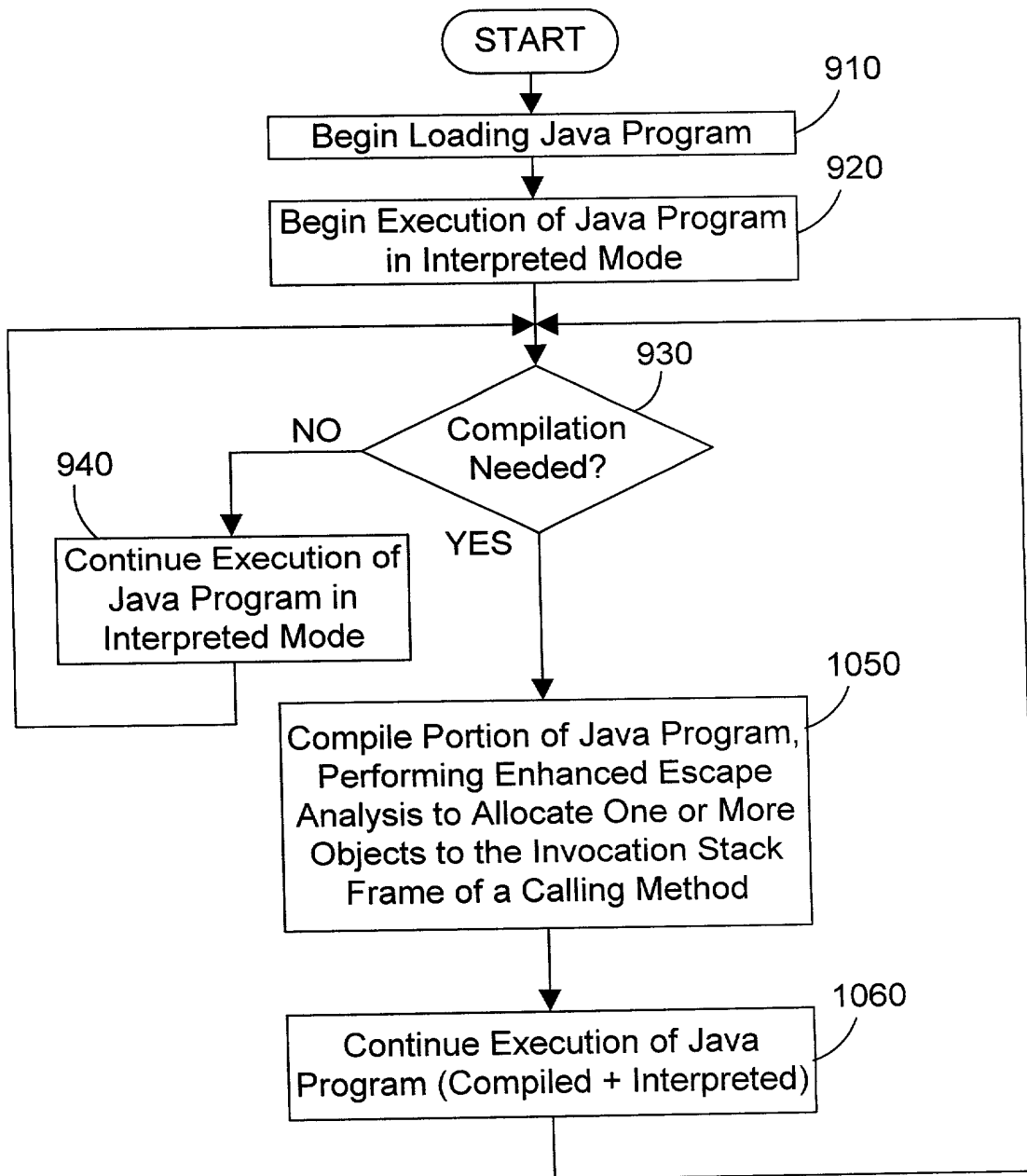


FIG. 10

FIG. 10

1050

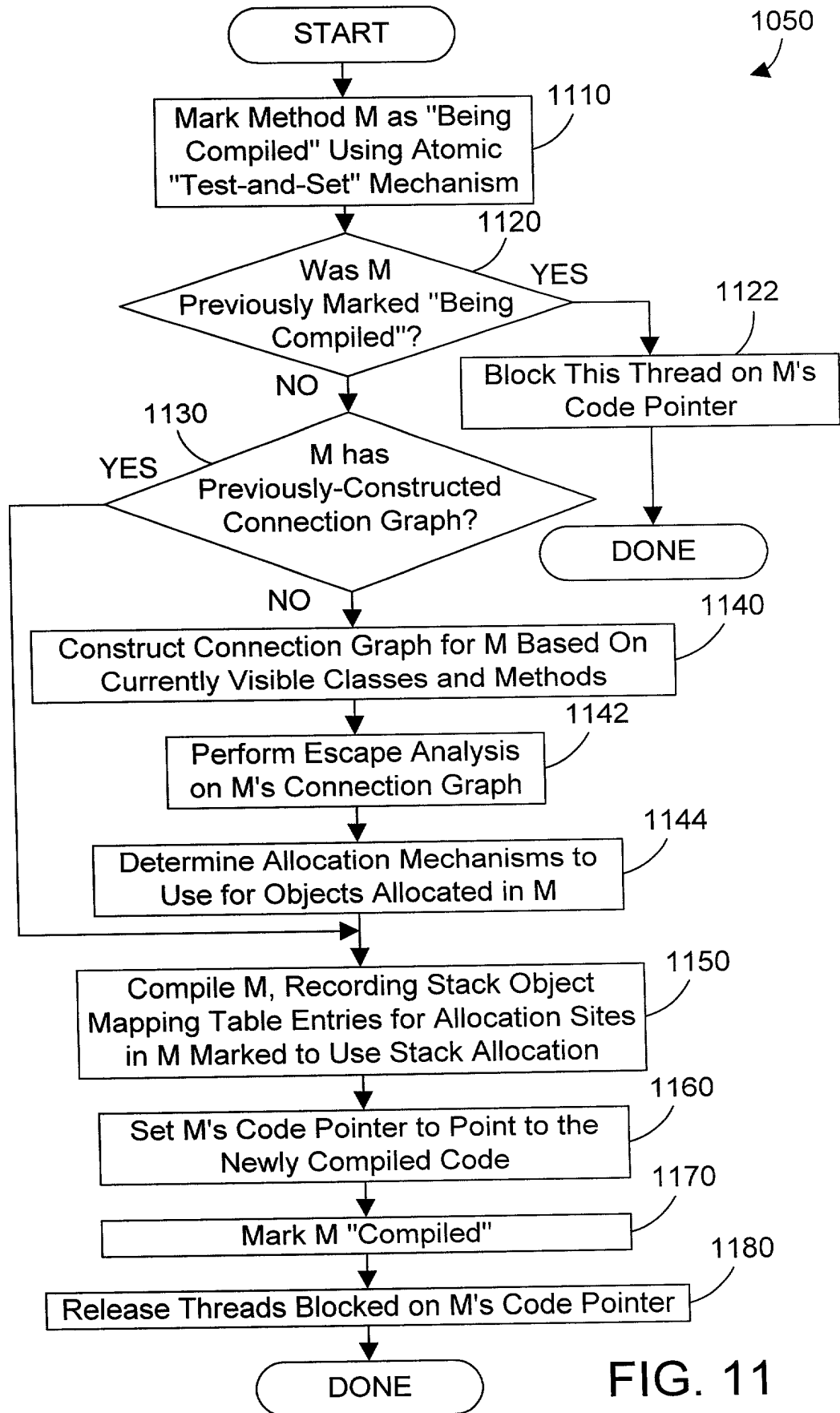


FIG. 11

FIG. 11

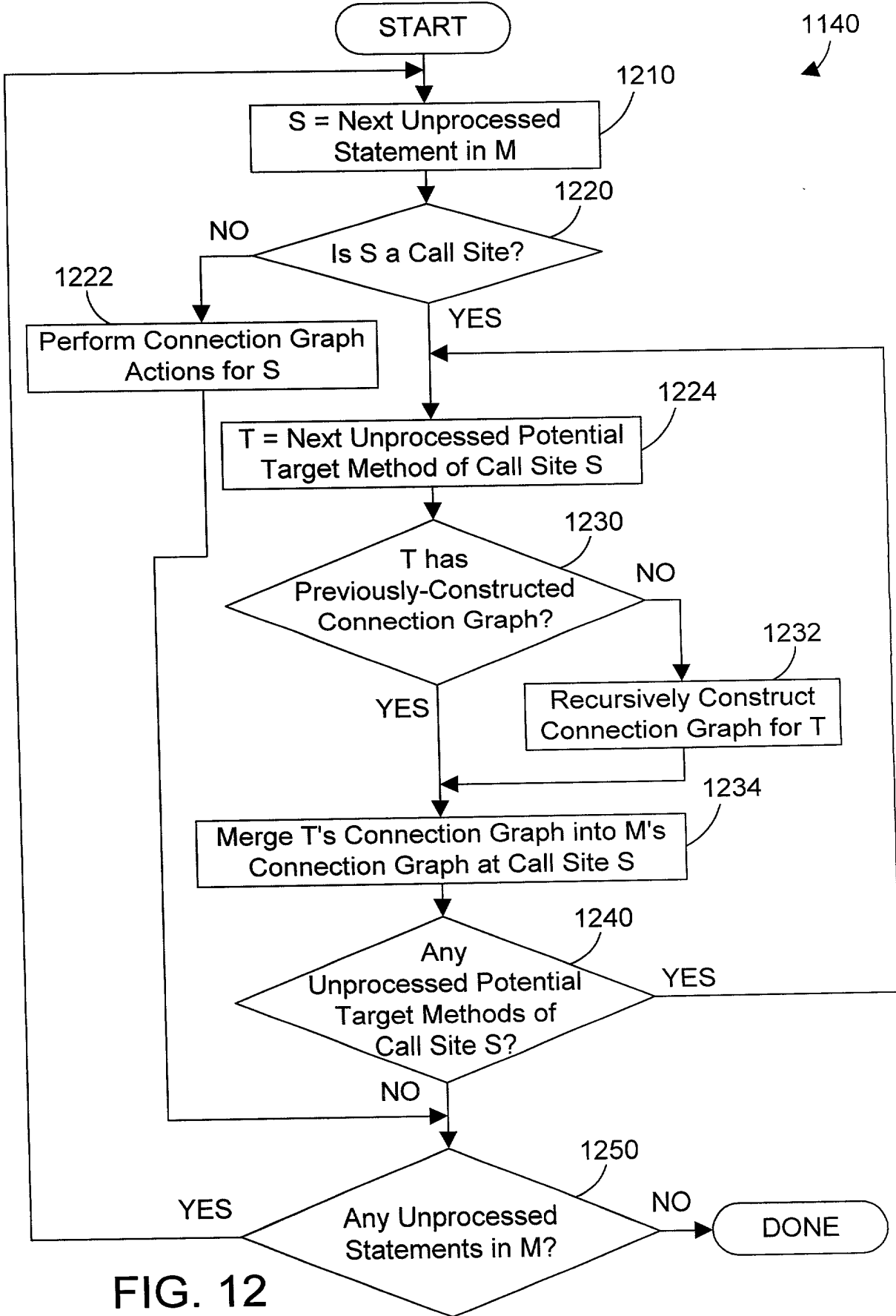


FIG. 12

FIG. 12

1060
↙

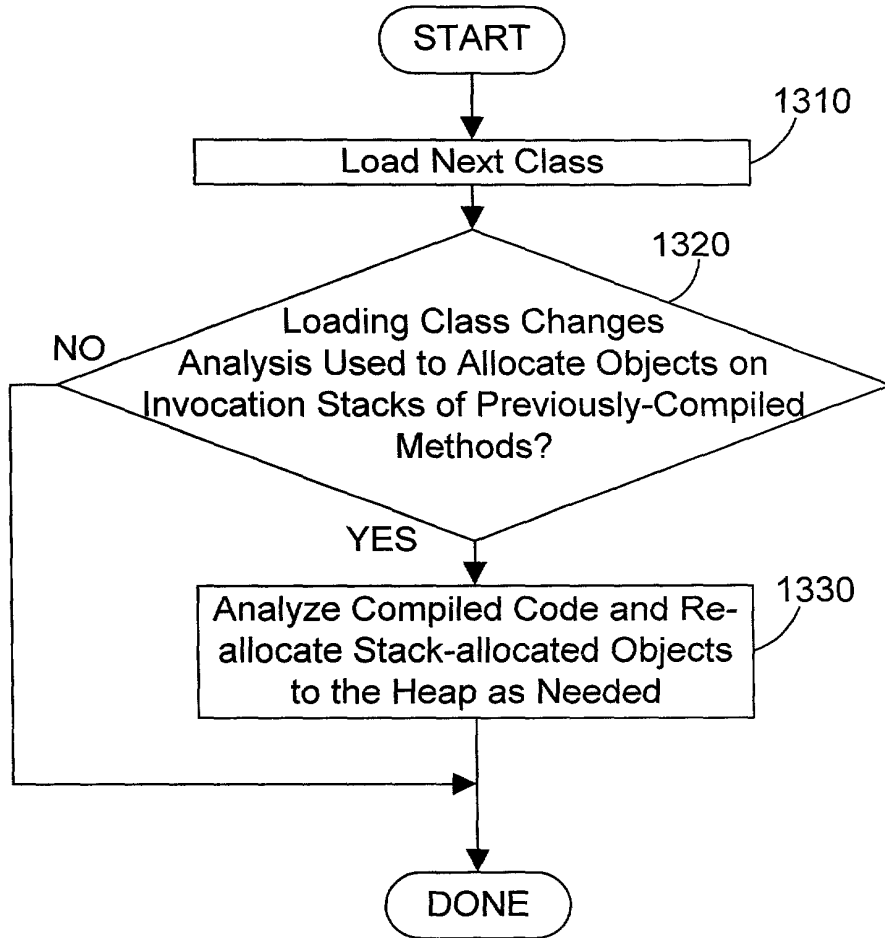


FIG. 13

FIG. 13

1060

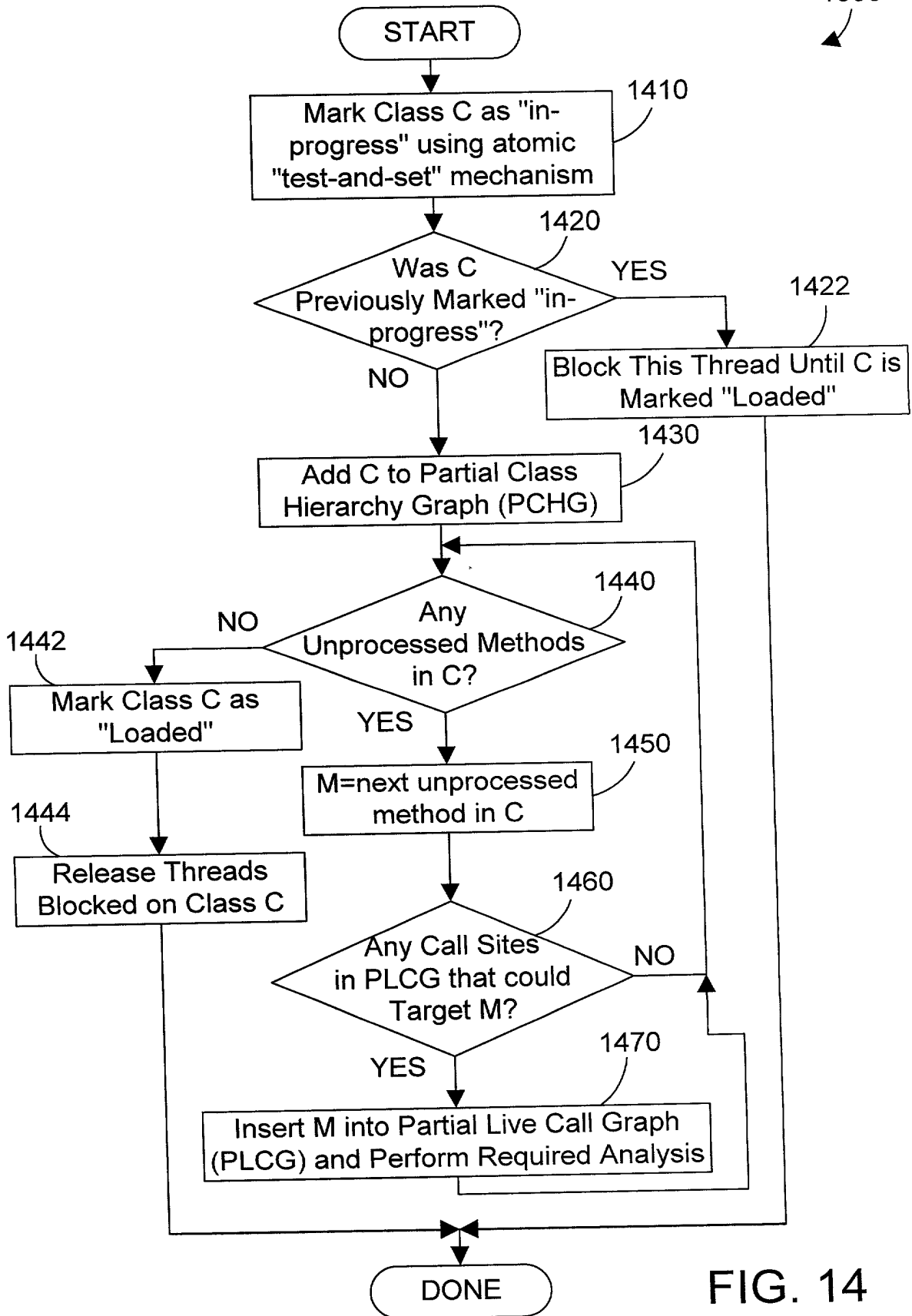


FIG. 14

FIG. 14

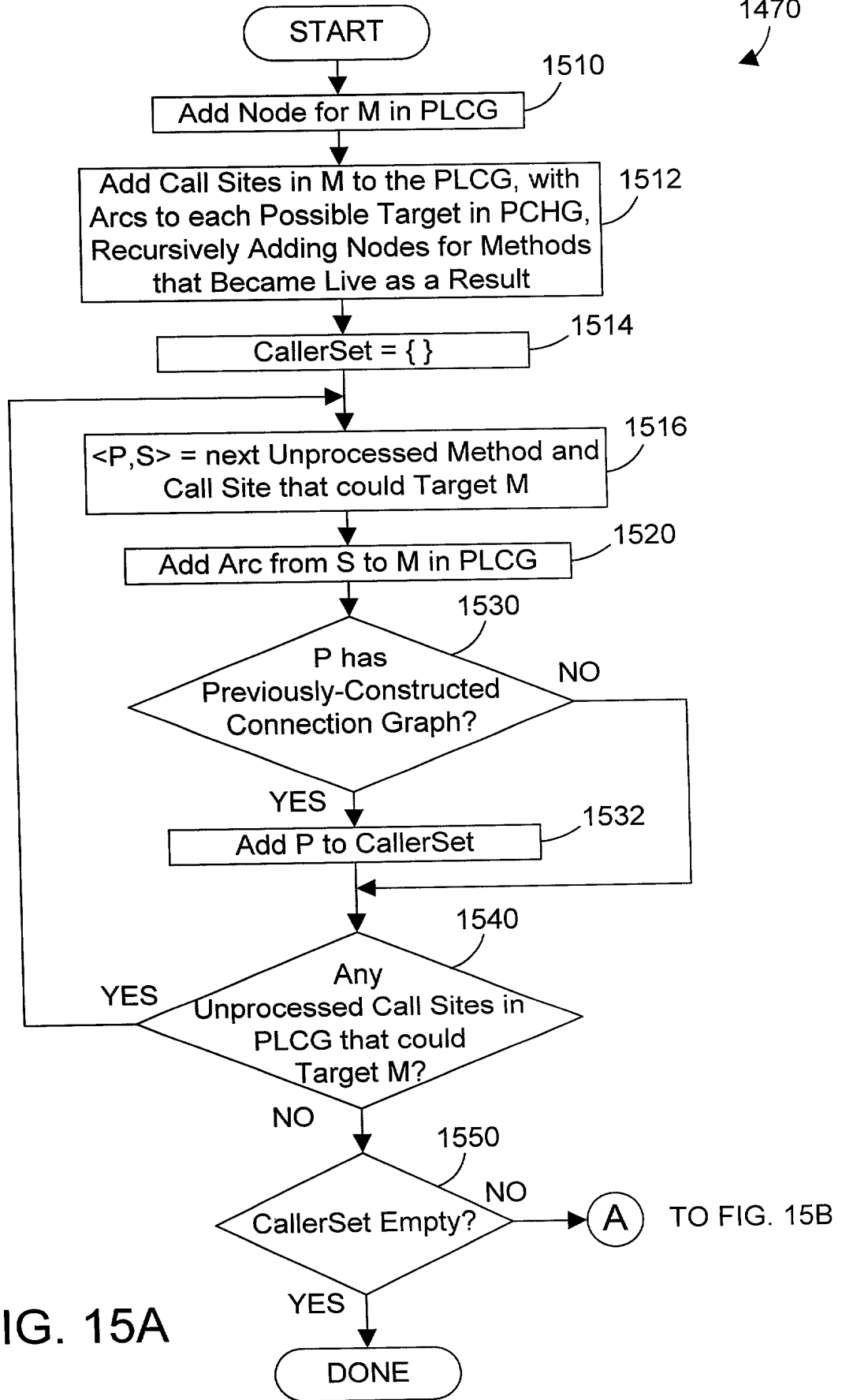


FIG. 15A

FIG. 15A

FROM FIG. 15A

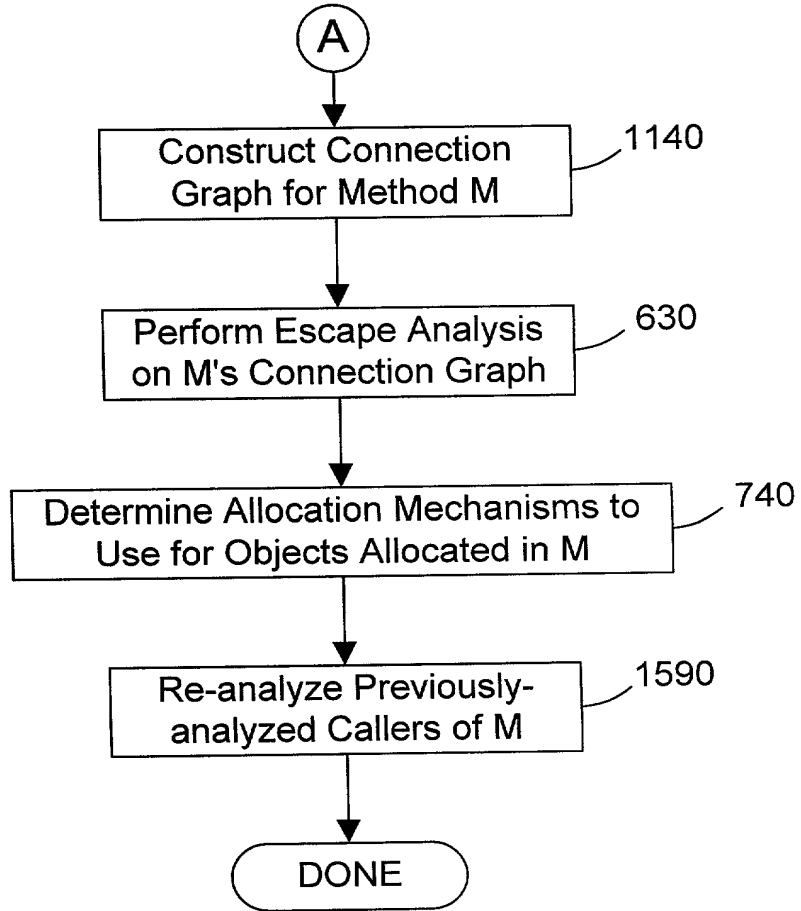


FIG. 15B

FIG. 15B

1590

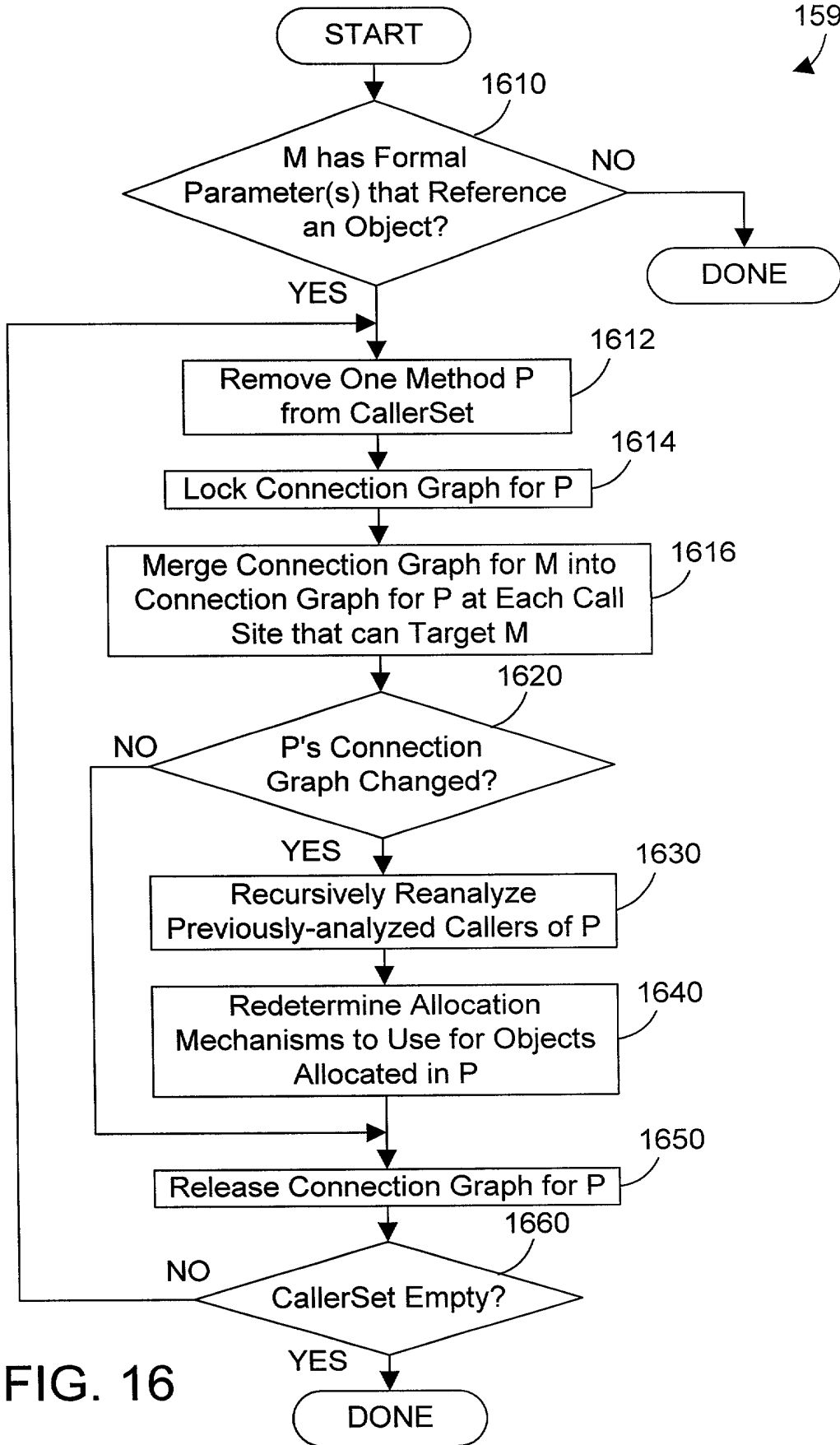


FIG. 16

FIG. 16

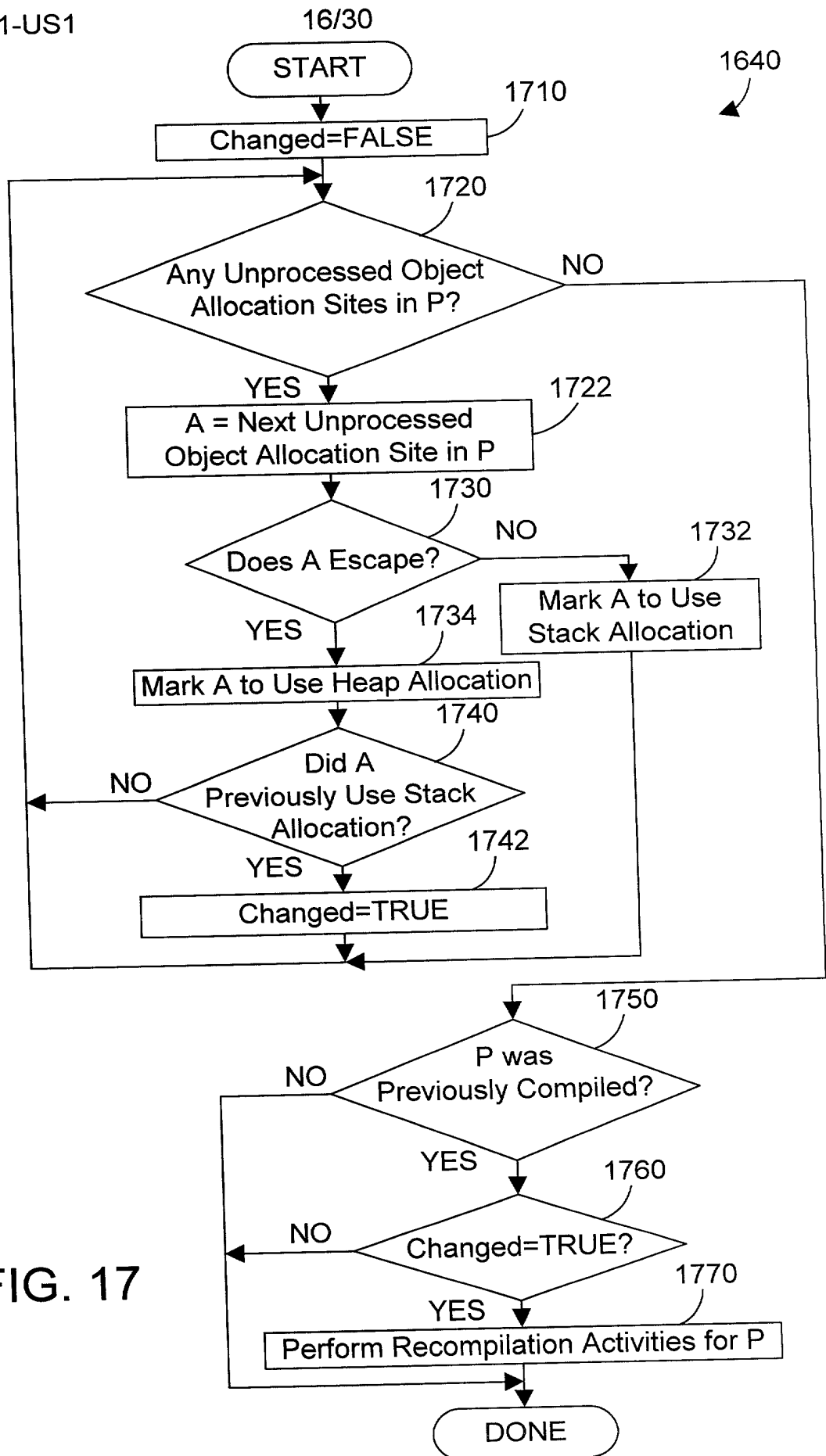


FIG. 17

1640

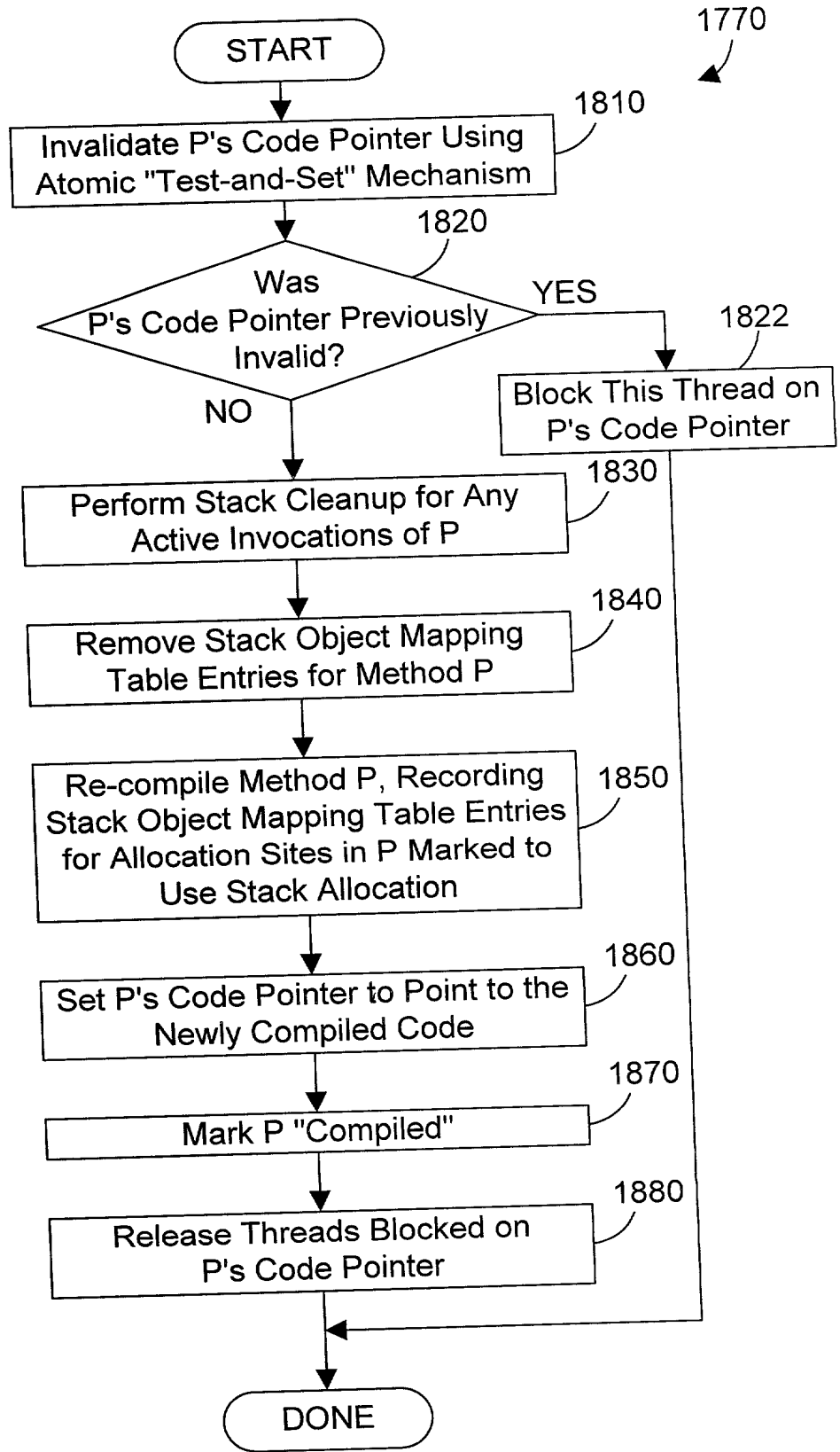


FIG. 18

FIG. 19

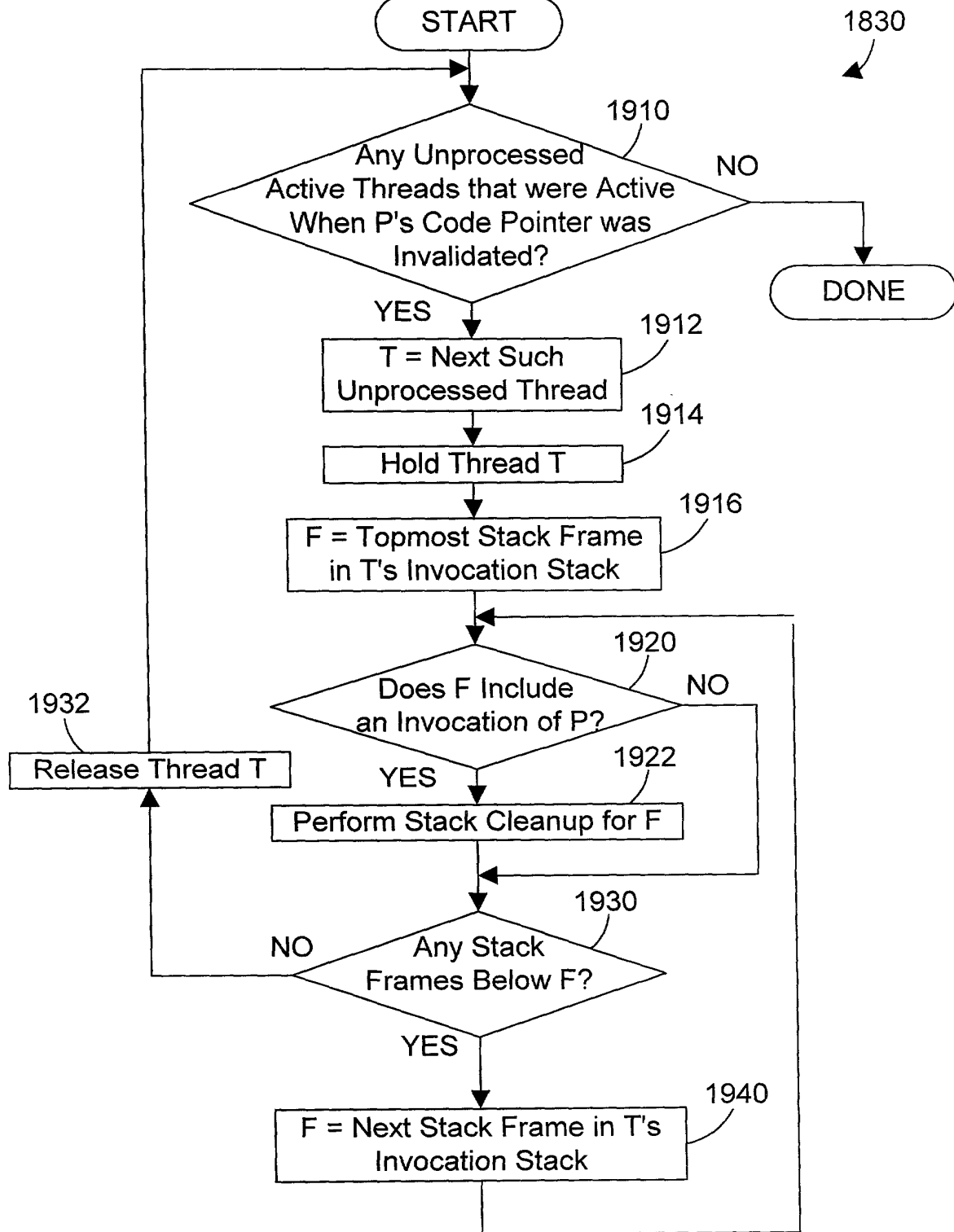


FIG. 19

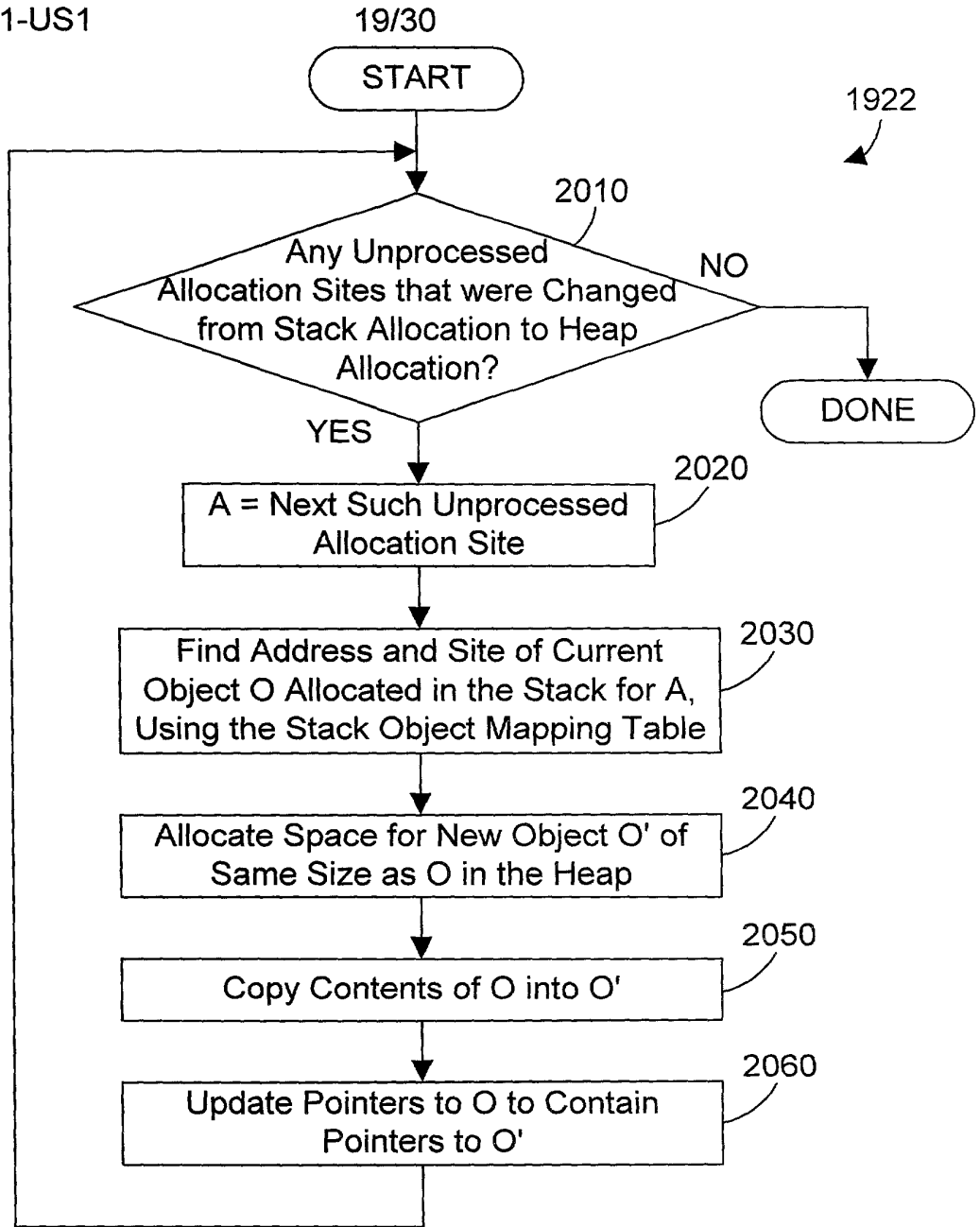


FIG. 20

FIG. 20

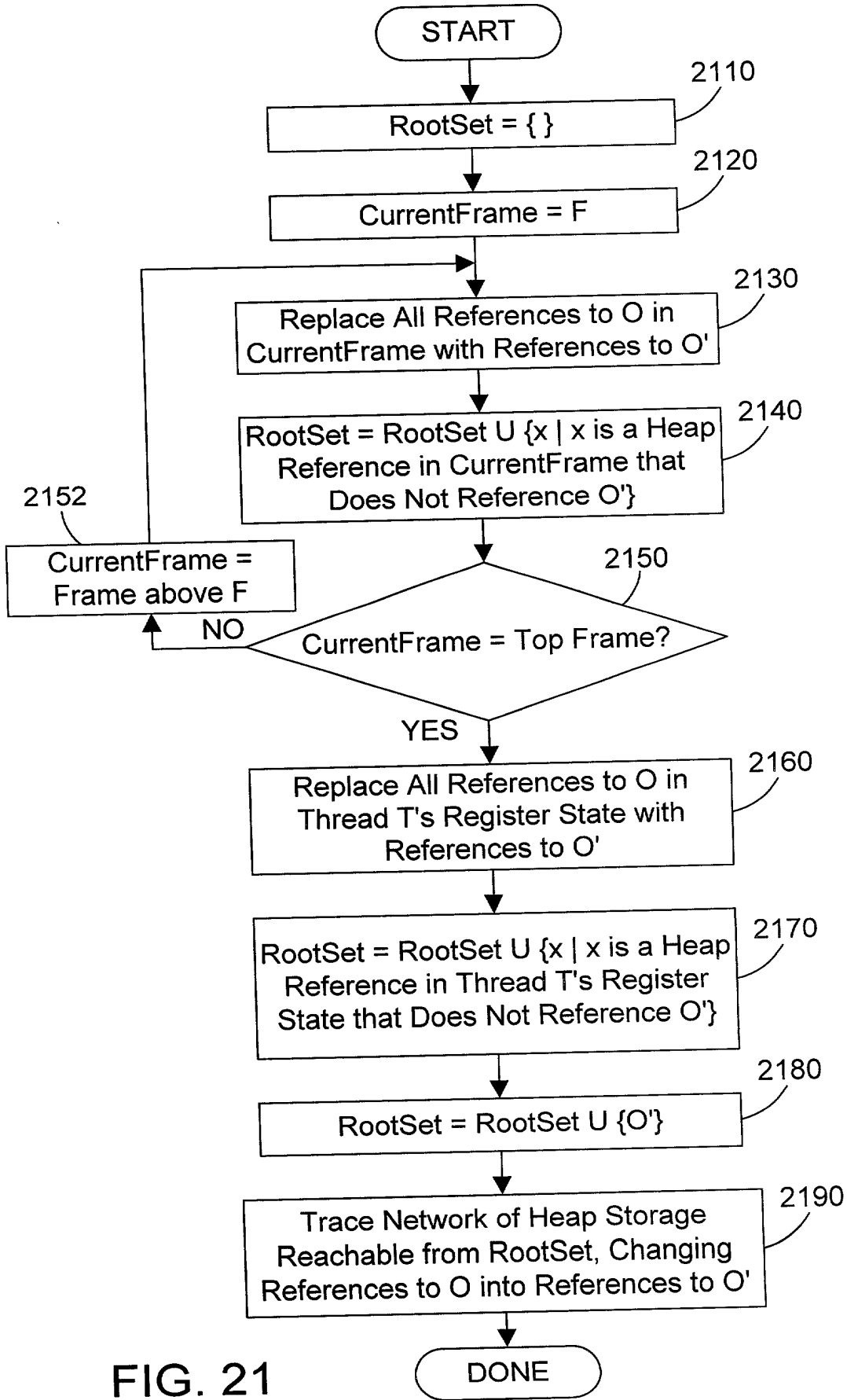


FIG. 21

FIG. 22

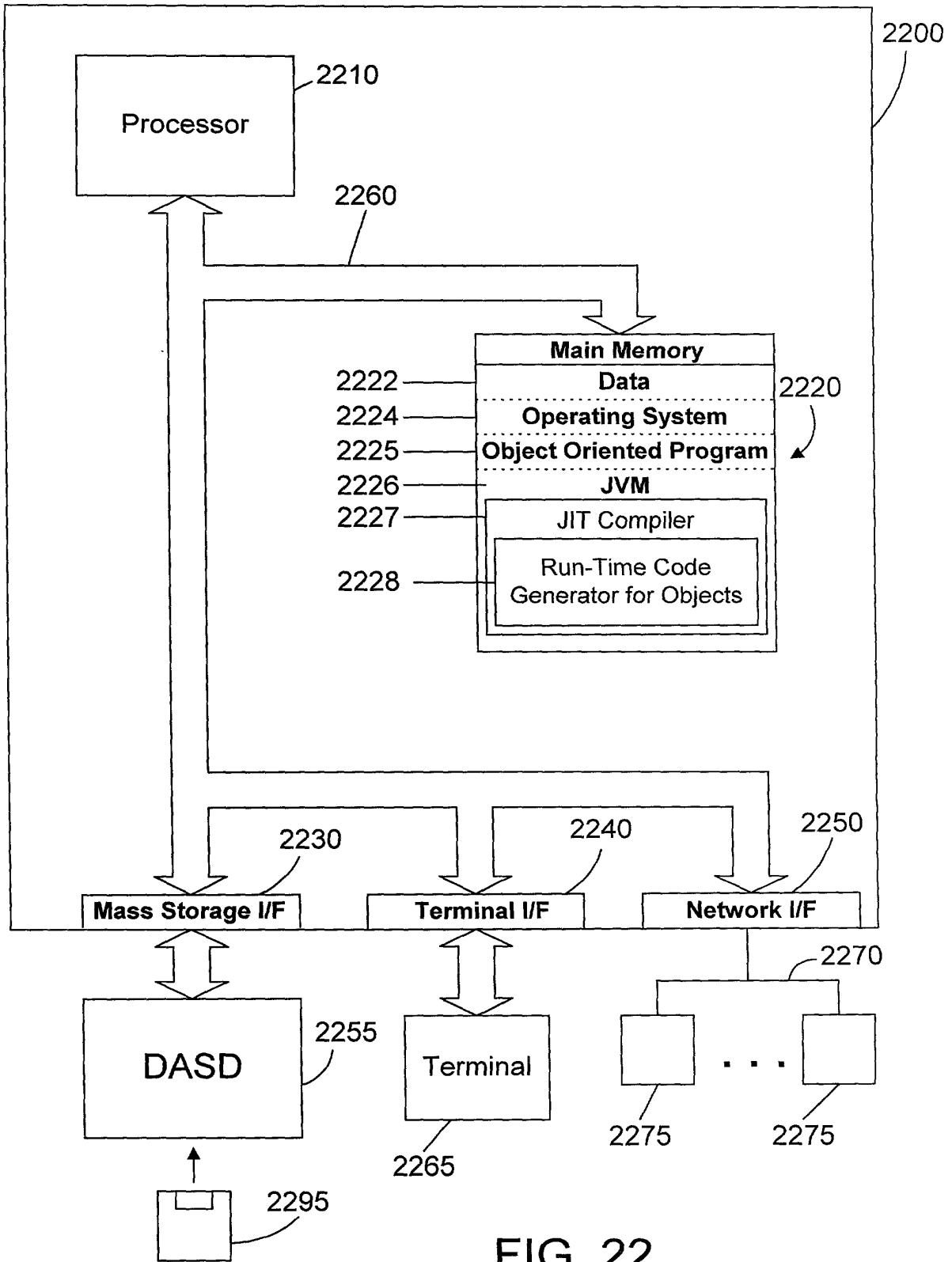


FIG. 22

```
class ComplexNumber {
    int r;
    int i;
    ComplexNumber(int a, int b) {r=a; i=b;}
    int realPart() {return r;}
    int imagPart() {return i;}
}
```

FIG. 23

```
class GeneralClass {
    static void examine(ComplexNumber c) {}
}
```

FIG. 24

```
class SpecificClass extends GeneralClass {
    static ComplexNumber savelt;
    static void examine(ComplexNumber c) {savelt = c;}
}
```

FIG. 25

```
class ExampleClass {
    GeneralClass gCls;
    static void exampleMethod(int a, int b) {
        ComplexNumber cn = new ComplexNumber(a,b); //A1
        doSomeWork(cn);
        gCls.examine(cn);
    }
    static void doSomeWork(ComplexNumber x) {
        if (x.imagPart() < 0) {
            Class specClass = Class.forName("SpecificClass");
            gCls = (GeneralClass)specClass.newInstance();
        } else {
            gCls = new GeneralClass(); //A2
        }
    }
}

public static void main(String[] argr) {
    int i, j;
    for (i=0;i<100;i++) {
        for (j=0;j<100;j++) {
            exampleMethod(i,j);
        }
    }
    exampleMethod(-1,-1);
}
}
```

FIG. 26

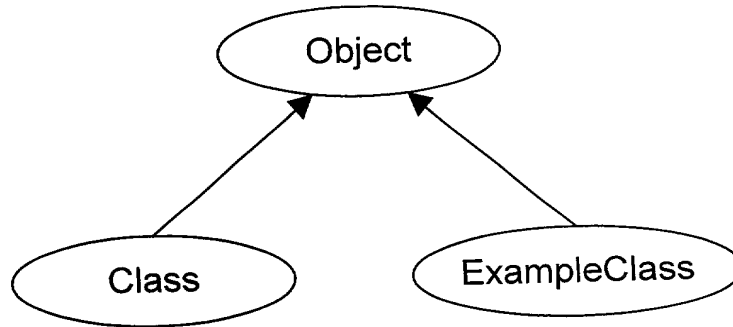


FIG. 27

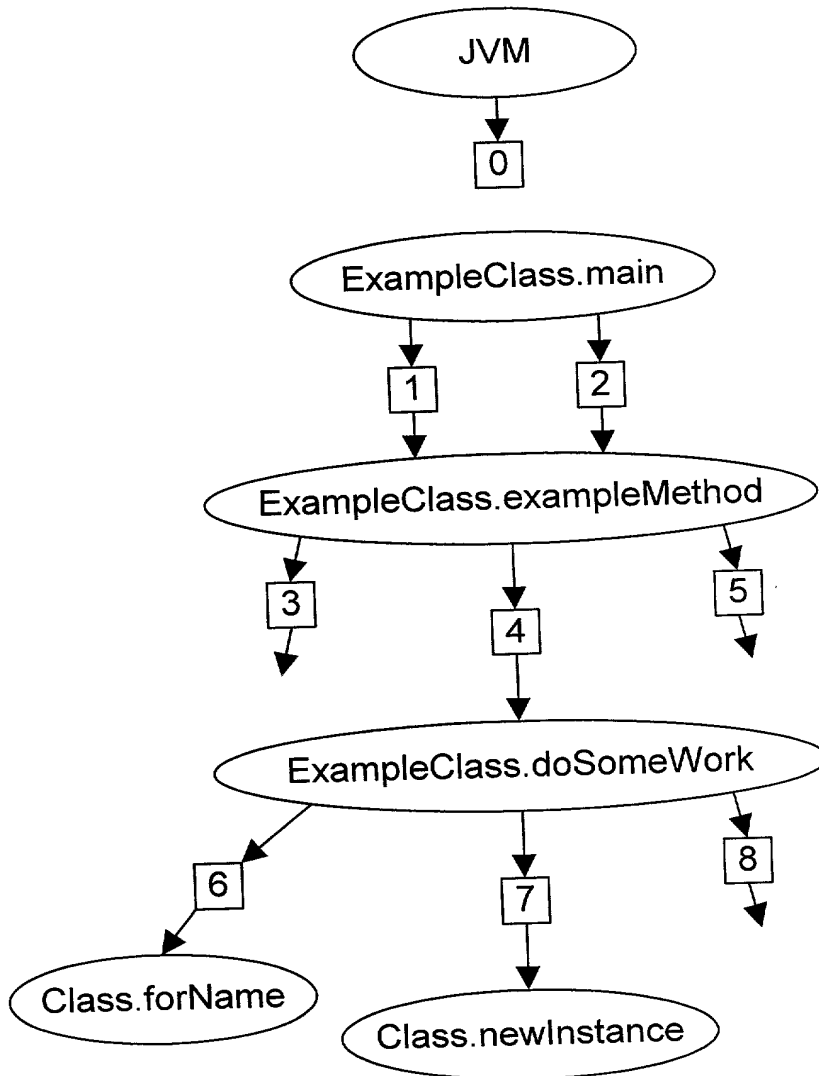


FIG. 28

FIG. 28

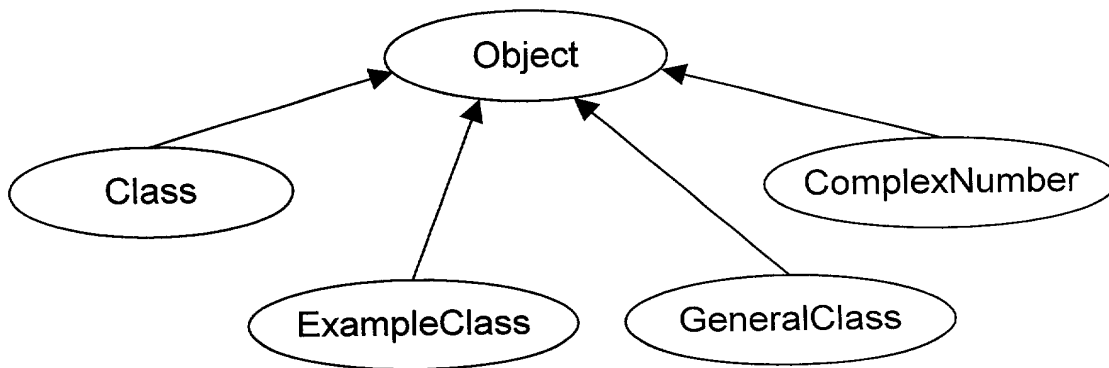


FIG. 29

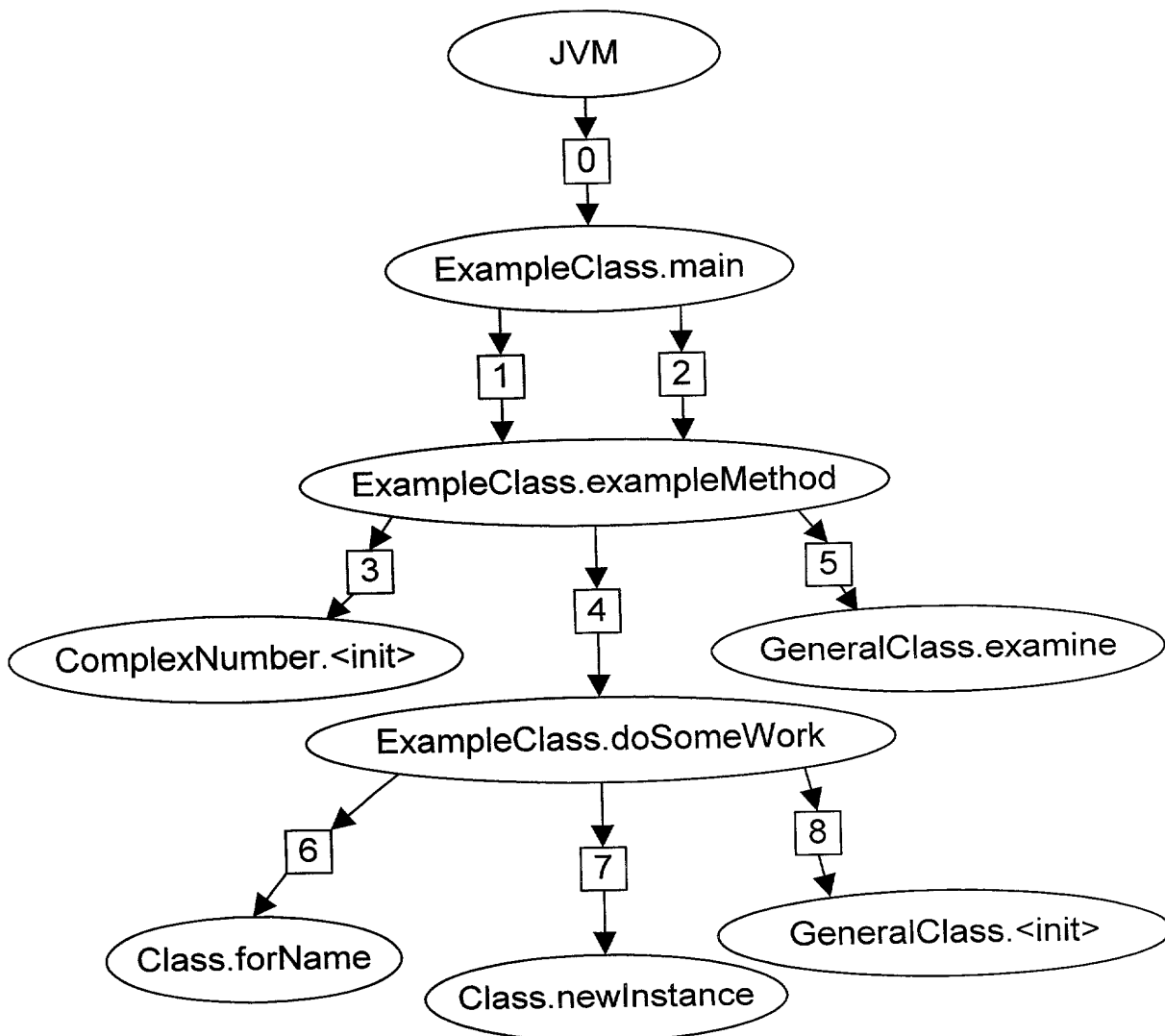
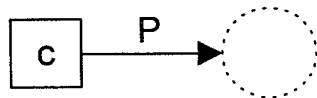


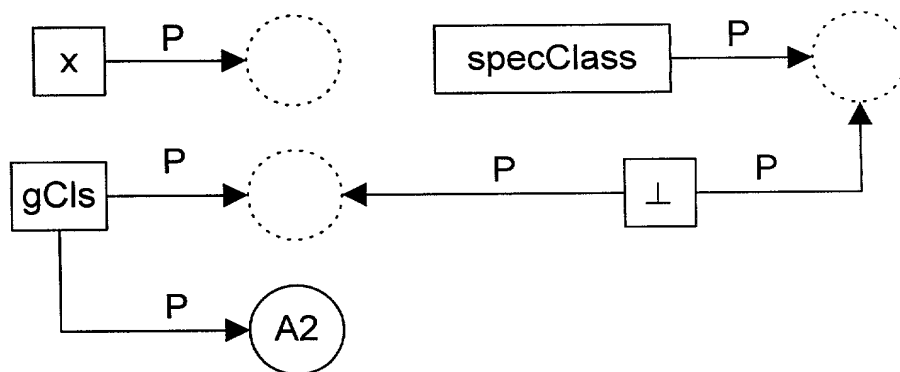
FIG. 30

FIG. 29



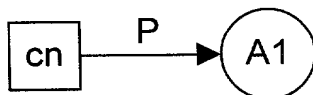
GeneralClass.examine()

FIG. 31



ExampleClass.doSomeWork()

FIG. 32



ExampleClass.exampleMethod()

FIG. 33

Site	Size	Offset
A1	8	64

FIG. 34

TABLE 1 OF 3

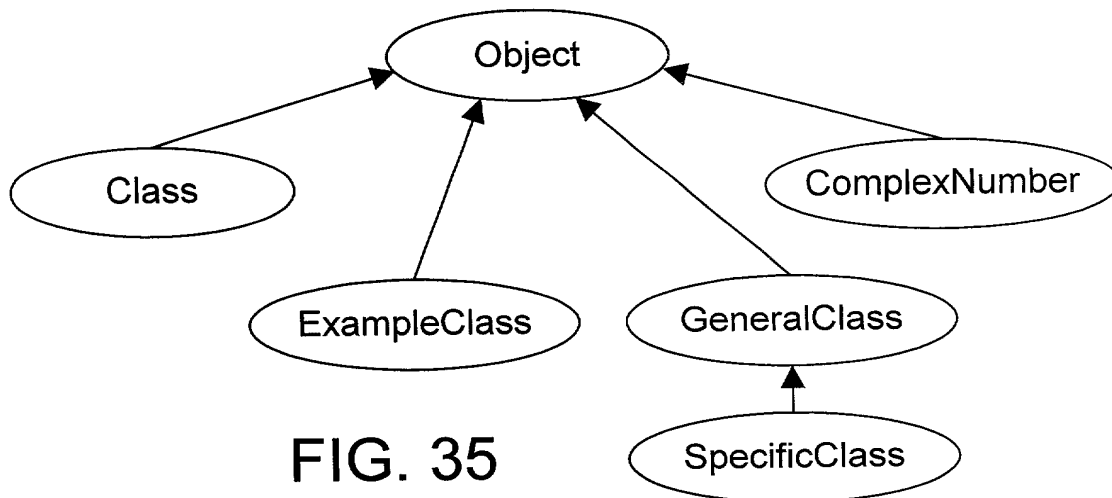
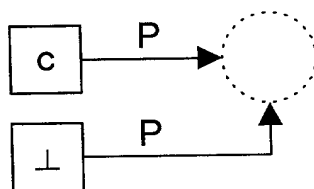
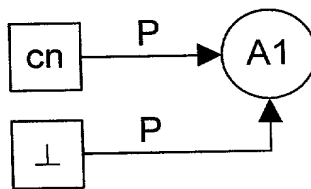


FIG. 35



GeneralClass.examine()

FIG. 36



ExampleClass.exampleMethod()

FIG. 37

FIG. 35

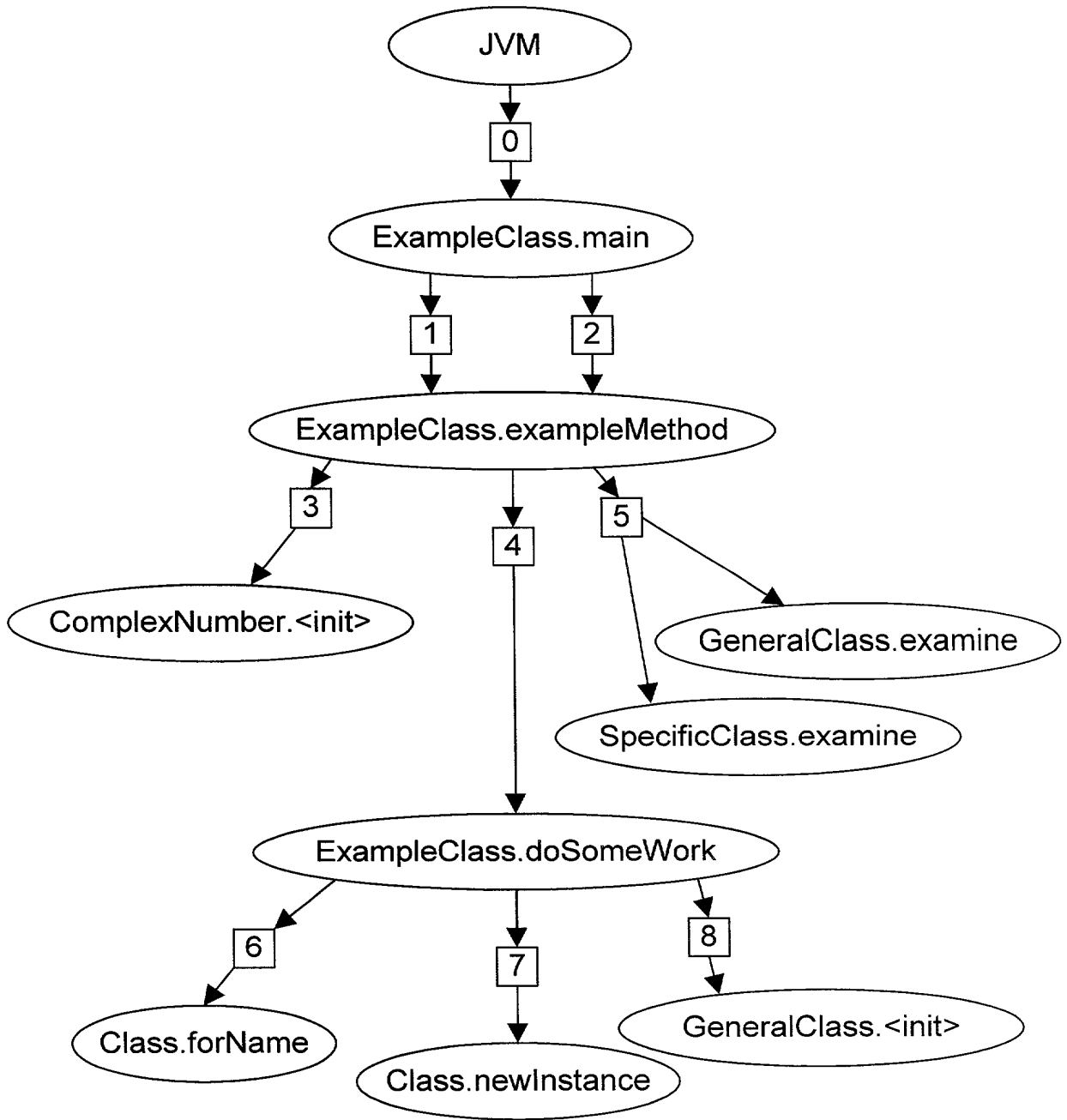


FIG. 38

FIG. 38

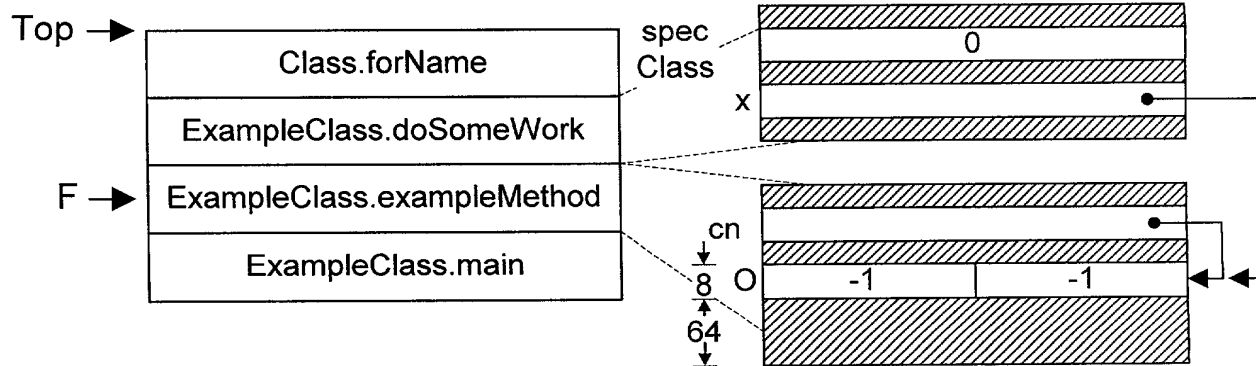


FIG. 39

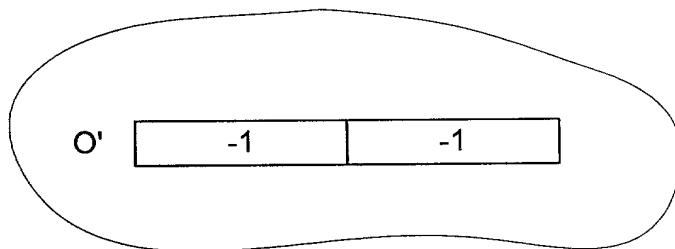


FIG. 40

FILED AT THE USPTO

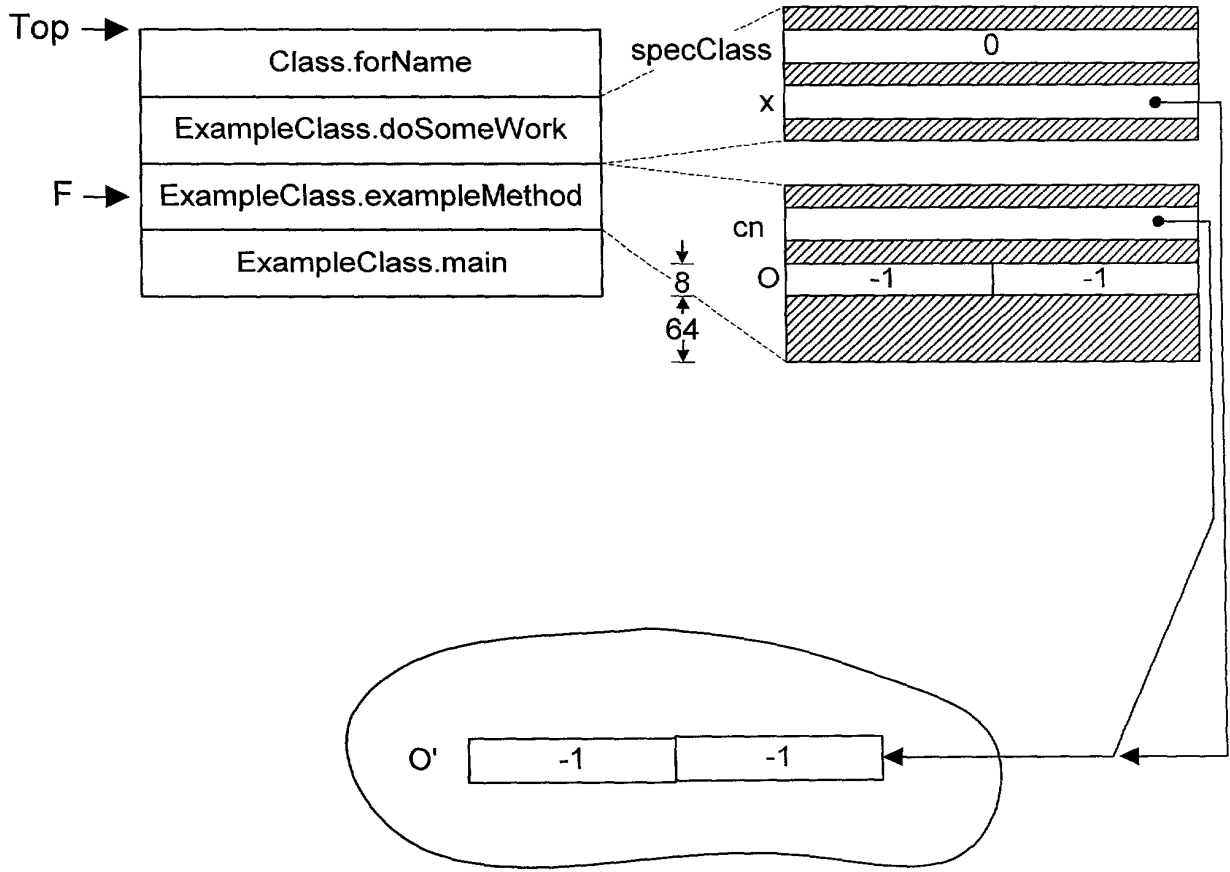


FIG. 41

FILED OCT 20 2000